

# **QMesh: A Synchronized, Flooded Mesh Network Protocol for Voice**

**Dan Fay, Ph.D. (KG5VBY)**

## **Abstract**

Wireless mesh networking protocols, such as APRS and Gotenna Mesh, allow for reliable communication between devices by relaying data between nodes until the data reaches its destination. While these mesh networks successfully communicate telemetry and small data bursts, they cannot provide the continuous streaming bandwidth needed to carry real-time voice communications. Presented here is QMesh, which leverages the FM capture effect in the LoRa waveform to allow for reliable synchronized, flooded mesh networking that can support low data rate digital voice communications. An overview of the theory, protocol design, experimental hardware, initial test results, and plans for QMesh will be discussed.

## **Overview of LoRa**

LoRa, which is a concatenation of “Long Range”, is Semtech’s proprietary Chirp Spread Spectrum (CSS) modulation. It is targeted as a low data rate, low-power, low-cost wireless communications waveform that is used for low-power sensing and other Internet-of-Things (IoT) uses. LoRa possesses two advantages over other, more commonly used waveforms such as Frequency Shift Keying (FSK) and On-Off Keying (OOK). The first advantage is that by spreading its energy across a large bandwidth (up to 500KHz), it can better resist interference from narrowband interference. Second, LoRa enjoys substantially better receive sensitivity than most other narrowband waveforms for a given bitrate.

Semtech provides several LoRa products: the SX127X series, the SX126X series, and the SX1280. The SX127X series is the original LoRa radio line designed for sub-GHz communications. The SX126X series is a successor to the SX127X series that enjoys lower receiver power consumption and can cover all frequencies from 150MHz to 960MHz. The SX1280 is a LoRa product designed to work on the 2.4GHz ISM band. It enjoys higher LoRa communication rates than the sub-GHz modems. There are also several products by other companies that combine a LoRa radio with a microcontroller. These include the Microchip SAM R34/R35 and the STM32WL series. It is of note that the STM32WL is likely the first LoRa modem die that is not fabricated by Semtech, as the SX1262 modem on it is on the same silicon die as the STM32L0 MCU (the SAMR34/35 package together an MCU die and an SX1276 die).

## **The LoRa Waveform**

LoRa is a form of Chirp Spread Spectrum (CSS). CSS modulation encodes information within chirps, typically with the goal of improved system gain and narrowband interference rejection. The purported benefits of the LoRa waveform are several. First, it enjoys a higher receive sensitivity for a given bitrate than do common FSK and OOK waveforms. Second, the spread spectrum characteristics of the waveform improve the resistance of the LoRa waveform to narrowband interference. These characteristics make it well-suited for low-power, battery-operated wireless sensing devices that operate on the unlicensed Industrial, Scientific, and Medical (ISM) bands like the 868MHz and 915MHz bands.

Unfortunately, Semtech does not provide detailed information about the LoRa waveform, so the information presented here comes from a combination of Semtech-provided documentation, reverse-

engineering projects, and Semtech patents. At a basic level, LoRa is a chirped  $m$ -ary FSK, where a large set of FSK tones are multiplied by a chirp. This operation encodes the information as a “break” in the chirp. A LoRa chirp is broken up into multiple, short-duration tones, known in the Semtech documentation as “chips”. They range from 32 up to 4096 chips per chirp, depending on the Spreading Factor (SF) setting. The number of chips per chirp is equal to  $2^{SF}$ .

There are three major LoRa settings: bandwidth (BW), spreading factor (SF), and coding rate (CR). On the sub-GHz LoRa products, bandwidths range from 500KHz all the way down to 7.8KHz, although the most commonly-used bandwidths are 500KHz, 250KHz, and 125KHz (NOTE: the 2.4GHz LoRa waveform uses wider bandwidths than the sub-GHz waveform). Narrower bandwidths trade higher receive sensitivity for lower bitrates. Spreading factors range from 5 to 12, with lower spreading factors trading a higher datarate for lower receive sensitivity. Finally, the coding rate specifies the Hamming code used to provide error detection/correction. Officially, the coding rate can be 4/5 (CR=1), 4/6 (CR=2), 4/7 (CR=3), and 4/8 (CR=4); however, there does exist an undocumented feature in at least some of the LoRa modems that allows for completely disabling the built-in error correction by setting CR=0.

LoRa does not tolerate much frequency drift within a given packet transmission, due to the close frequency spacing between chips. There is a Low Datarate setting that trades two bits per symbol in exchange for a sixteen-fold improvement in frequency drift tolerance. Regardless, while it is not necessary for higher datarates, a TCXO is necessary for lower datarates (less than BW=62.5KHz/SF=12), and long transmission times ( $> \sim 3$  seconds). A TCXO is also recommended for transmit power over 20dBm due to heating of the oscillator by the power amplifier (PA), unless the board is carefully designed to thermally isolate the oscillator from the modem chip. Semtech’s LoRa modems, however, are highly tolerant of absolute frequency error, and can successfully receive LoRa packets whose center frequencies are off by up to  $\pm 10$ -25%.

### **LoRa Implementation**

LoRa packets have a preamble to enable the receiver to detect and synchronize the receiver, as well as a “sync word” which appears to be a reverse chirp that may also synchronize the receiver. Different sync words can be specified; the LoRa receiver will ignore other sync words that it is not set to. Typically, two different sync words are used: 0x12 for private networks and 0x34 for public networks. While other sync words can be set, they should be thoroughly tested, as some sync words are not reliably received. Note that it is possible to set sync words that are incompatible between the SX127X series and the SX126X series.

Another feature provided by Semtech’s LoRa implementations is the ability to reverse the I and Q samples in the receiver. This has the effect of turning upchirps into downchirps, and vice versa. The practical use for this is that forward and reverse chirps are orthogonal and largely non-interfering with each other, which facilitates full-duplex communications. LoRaWAN, for example, uses different IQ polarities on the uplink and downlink to make it possible to simultaneously transmit and receive, although real-world LoRaWAN gateways do not appear to perform full-duplex communications.

Semtech provides two different packet modes: “implicit header” and “explicit header”. Implicit header mode is a standard fixed-length packet. Explicit header mode provides a quasi-link layer with a header that describes the length of the data payload, the coding rate of the payload, is protected by a header CRC. LoRa chipsets can also automatically compute the CRC of a packet.

Semtech implements the LoRa waveform on several product lines. The original one is the SX127X series, which is the original LoRa chipset for sub-GHz frequencies. There is also the newer SX126X series, which is another sub-GHz chipset with some improved features such as lower receiver power consumption, higher transmit power (22dBm on the SX1262 vs. 20dBm on the SX1276) on some models, as well as support for SF=5. Finally, the SX1280 is a LoRa chipset for 2.4GHz that uses a wider bandwidth for higher data rates.

### **Overview of Wireless Mesh Networking**

Mesh networking is a non-hierarchical network topology where nodes work cooperatively to move data packets through the network. Typically, mesh networks can handle nodes entering and leaving the network while still maintaining a functioning network. Wireless mesh networks are a common use case for mesh networking, particularly for Mobile Ad-hoc NETWORKs, or MANETs. MANETs are highly ad-hoc networks where the network topology is expected to dynamically change: nodes can enter or leave the network on a whim, nodes can change location, etc. Real-world examples of MANETs include e.g. groups of hikers sharing telemetry data, vehicle-to-vehicle communications, and drone swarms.



Source: <https://blog.particle.io/2018/04/28/how-to-build-a-wireless-mesh-network/>

Figure 1: Diagram of a mesh network.

Mesh network protocols fit into two broad categories: flooded and routed protocols. Flooded protocols are where every node repeats every packet that it receives. Flooded protocols enjoy the benefits of simplicity (no need to generate and maintain routes) and self-healing at the expense of efficiency (lots of unnecessary retransmissions of packets). Routed protocols, where nodes only repeat packets if they are along a defined route, are more efficient, but more challenging to build and maintain working routes as network conditions change.

Another issue with mesh network protocols, particularly ones that use a shared medium such as RF communications, is known as the *broadcast storm problem*. When multiple nodes transmit at the same time, their packets collide. If the collisions are completely destructive, the wireless communication will fail, requiring a retransmit. Multiple retransmits can in turn destructively collide, causing a situation

where endless collisions heavily degrade the wireless channel. Mitigations for this problem include using Time Division Multiple Access (TDMA) slotting (such as “Slotted ALOHA”) and Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA).

An alternative technique that is explored in this paper is what is known as Synchronized Flooding. In synchronized flooding, all nodes repeat a received packet at the *exact same time* (or at approximately the same time). While this guarantees that all the packets will collide, different physical-layer techniques can be used to reduce the likelihood of destructive collisions. Since all retransmitted packets contain the same data, a node only needs to receive one of the retransmitted packets to receive all of the data. An additional benefit of synchronized flooding is that it provides isochronous bandwidth for streaming applications such as real-time voice communications.

Examples of physical-layer techniques to increase the likelihood of successful packet receipt include:

1. **Communicating simultaneously on multiple channels.** A specific case of this is Frequency-Hopping Spread Spectrum-Multiple Access (FHSS-MA). FHSS-MA uses multiple, orthogonal hopping patterns across different nodes to reduce the probability of collisions.
2. **Code Division Multiple Access (CDMA).** Use different pseudo-noise (PN) codes with different nodes while transmitting with Direct Sequence Spread Spectrum (DSSS).
3. **Capture effect.** Many receivers will receive a stronger signal and ignore a weaker signal. This effect is well-known with constant-envelope modulations, where it is called the “FM capture effect”.

The capture effect describes when a receiver, in the presence of two overlapping signals, receives exclusively the stronger of the two signals (i.e., “captures” it). A well-known version of this effect is known as the FM capture effect, as constant-envelope receivers generally experience it. This effect can be seen with FM radio in a marginal reception area. When there are two stations in the area, either one or the other station is received. This contrasts with amplitude modulation receivers (e.g., AM or SSB), where overlapping signals add together. The capture effect is important for any Industrial-Scientific-Medical (ISM) band device, as these devices need to be able to operate in an uncontrolled RF environment where other like devices may be communicating as well. As a constant-envelope modulation, LoRa receivers exhibit the capture effect, and as described in subsequent sections, the LoRa modulation exhibits properties that can be leveraged to increase the likelihood of successful capture.

Other examples of synchronized flooding exist in the literature. One key example is Glossy<sup>1</sup>, which demonstrated that successful capture was possible for the DSSS used in the physical layer of 802.15.4 transceivers. One caveat, however, was that the nodes must stay synchronized within one chip period (0.5 $\mu$ s in the case of the 250Kbps DSSS-based datarate used). Given that radio waves can travel roughly 150m in that time period, it is easy for wide-ranging networks to lose synchronization. While the situation improves at a lower symbol rate (say 4800baud), a 4x chip rate still has a chip period of roughly 52 $\mu$ s, or the time period it takes radio waves to travel roughly 15km.

---

<sup>1</sup> Ferrari, Federico & Zimmerling, Marco & Thiele, L. & Saukh, Olga. (2011). Efficient network flooding and time synchronization with Glossy. Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN'11. 73 - 84.

Later work<sup>2 3</sup> showed that LoRa transmissions enjoy similar capture benefits to Glossy for the LoRa waveform, provided that colliding LoRa symbols do not completely overlap with each other. In these papers, separation leading to successful capture was provided by a combination of small frequency errors in the modems' oscillators as well as an induced, random timing offset of up to half of a symbol period. A major benefit for LoRa for this application is the comparatively low symbol rate relative to the chip rate of DSSS. For a bitrate comparable to 4800bps, LoRa uses a symbol rate of 1ms, or roughly 19x the period of the DSSS-chipped 4800bps example mentioned previously. In a separate paper, the authors showed that non-spread spectrum, FSK-based synchronized flooding suffers from significantly lower capture rates, although these rates are improved when an interleaved FEC algorithm is used<sup>4</sup>.

Barrage Relay Networks<sup>5</sup> are another synchronized, flooded network protocol. It appears to be a higher performance waveform/protocol than either LoRa or 802.15.4. It also appears to split up payloads into multiple FEC-coded blocks. Each block is phase-dithered to reduce the likelihood that it constructively interferes with other simultaneous transmissions. Barrage Relay Networks also appear to have more sophisticated receiver processing, with some sort of combining algorithm used to leverage the multiple, colliding signals.

### **Overview of the QMesh Protocol**

The QMesh protocol is a synchronized, flooded mesh protocol that leverages the characteristics of the LoRa waveform to enable successful capture of one of the colliding packets. QMesh adds forward error correction to the LoRa PHY to improve the likelihood of successfully capturing colliding packets. The goal of this network protocol is to support voice communications over this mesh network via `codec2`-encoded voice. By enabling every QMesh-supporting device to serve as a repeater, better coverage can be obtained for amateur voice in ad-hoc situations such as special event communications and emergencies. Moreover, the single-frequency, store-and-forward architecture of the QMesh protocol, combined with the coding gain provided from high-performance forward error correction allows for compact, low-power (i.e. solar-powered), low-cost repeaters.

LoRa has two potential ways to "spread out" transmissions in order to increase the likelihood of successful capture. The first technique involves spreading out transmit energy at the chip level. Given that LoRa chirps are not continuous frequency sweeps, but rather a series of discrete chirps, slightly shifting the frequencies of the chips should reduce their interference between them. Likewise, by shifting the chirps in frequency and/or time, it should be similarly possible to reduce the amount of interference between colliding packets and increase the likelihood of successful capture. Figures 2 and 3 show how frequency and time shift reduce overlap between transmissions and increase the likelihood of successful capture.

---

<sup>2</sup> Hou, Y., Liu, Z., & Sun, D. (2020). A novel MAC protocol exploiting concurrent transmissions for massive LoRa connectivity. *Journal of Communications and Networks*, 22, 108-117.

<sup>3</sup> Liao, C., Zhu, G., Kuwabara, D., Suzuki, M., & Morikawa, H. (2017). Multi-Hop LoRa Networks Enabled by Concurrent Transmission. *IEEE Access*, 5, 21430-21446.

<sup>4</sup> Chun-Hao Liao, Makoto Suzuki, and Hiroyuki Morikawa. 2016. Toward Robust Concurrent Transmission for sub-GHz Non-DSSS Communication: Poster Abstract. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM (SenSys '16)*. Association for Computing Machinery, New York, NY, USA, 354–355. DOI:<https://doi.org/10.1145/2994551.2996703>

<sup>5</sup> Halford, T.R., & Chugg, K.M. (2010). Barrage Relay Networks. *2010 Information Theory and Applications Workshop (ITA)*, 1-8.

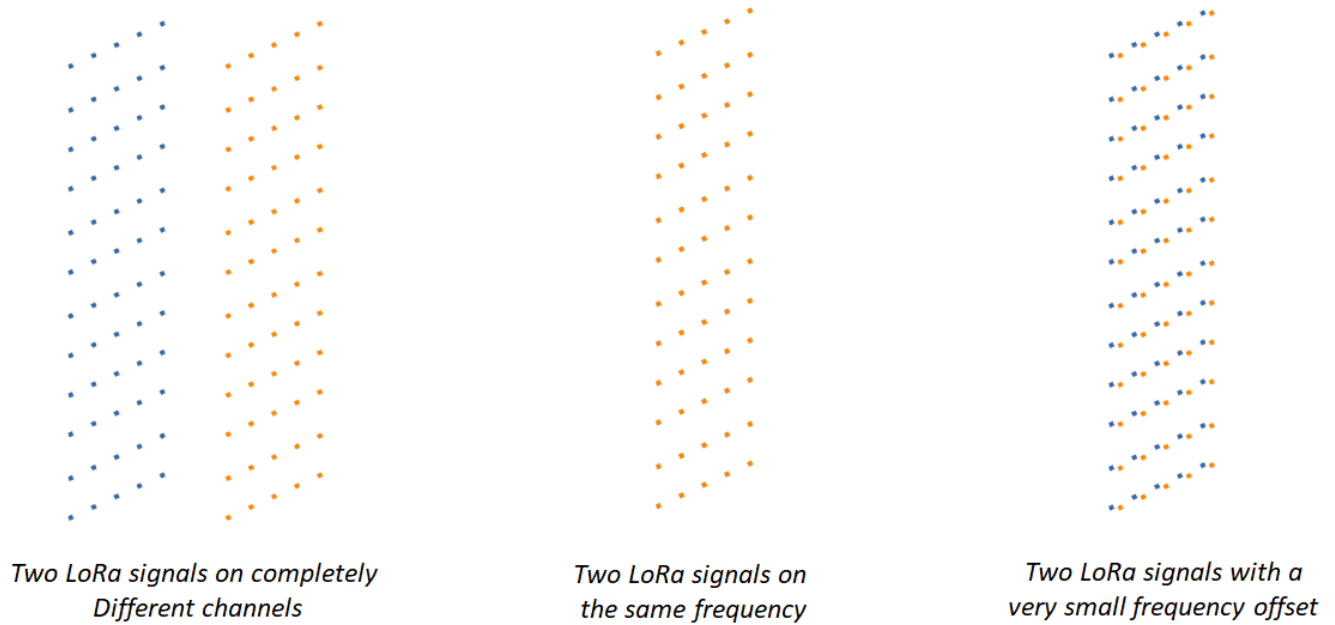


Figure 2: Overview of fine-grained frequency offset.

Another way to increase the likelihood of successful capture is to separate the chirps themselves, which is accomplished by randomly spreading out the chirps by larger amounts in frequency and/or time.

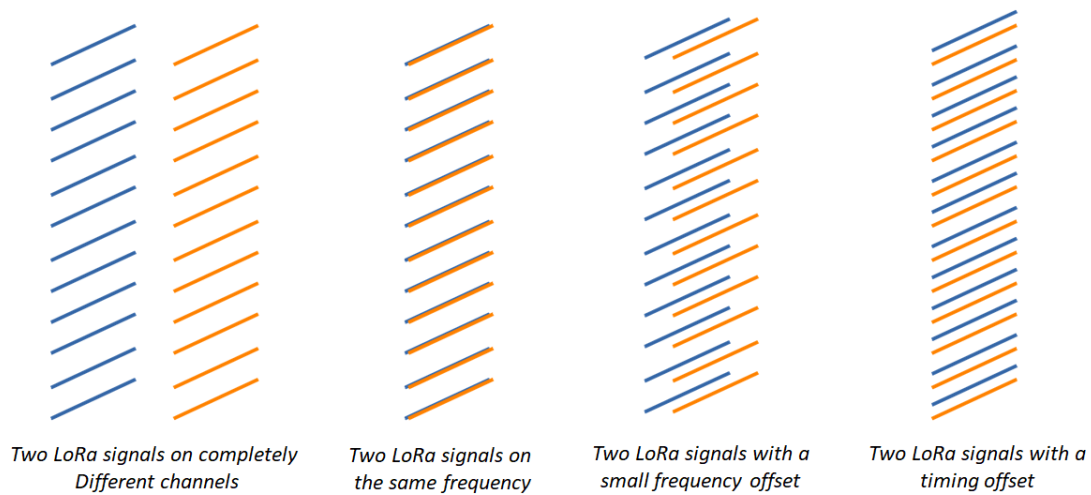


Figure 3: Coarse-grained frequency offset and timing offset.

QMesh's link-layer network protocol is a TDMA protocol that is based on equal-length timeslots, as shown in Figure 4. Within every third timeslot, a node can originate a transmission. Adjacent nodes that receive this transmission wait one timeslot, and then retransmit the packet. This procedure continues down the line until there are no more connected nodes that have not received the packet.

Such a setup provides several benefits. First, having a gap timeslot provides time for the system to receive a packet, decode it (some FEC algorithms can require significant time to decode), and prepare for retransmission. Second, this protocol eliminates interference from adjacent downstream transmissions. Finally, this protocol allows nodes a "second chance" to receive a missed transmission by receiving the transmission of a downstream retransmission.

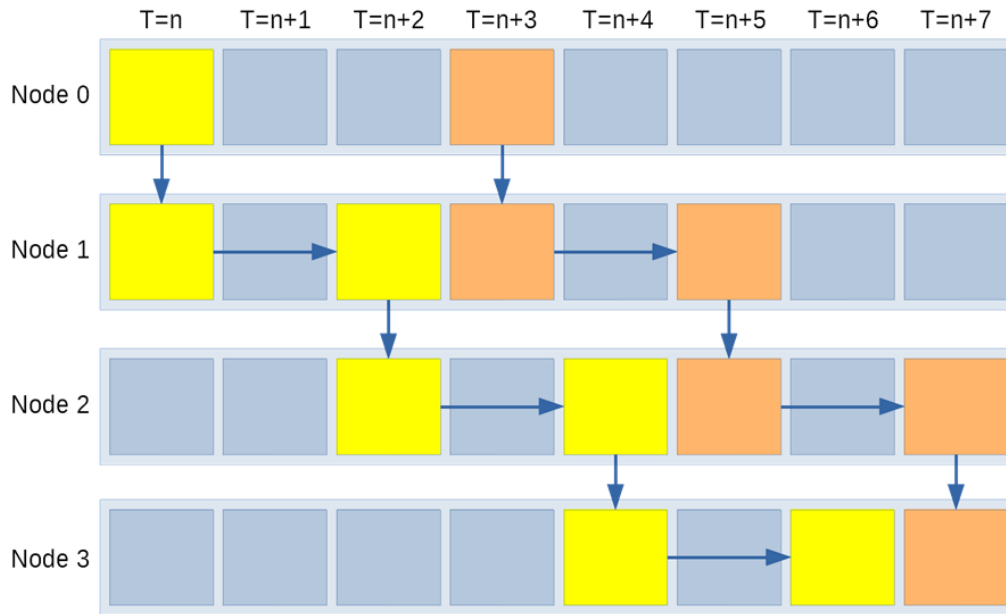


Figure 4: Overview of QMesh's TDMA protocol.

## Forward Error Correction

QMesh eschews LoRa's Hamming Code-based FEC in favor of its own, more-sophisticated FEC by using an undocumented feature in the SX126X that allows for completely disabling the built-in FEC by setting the CR=0. This capability was discovered in the documentation for the STM32WL<sup>6</sup>, which combines an STM32 microcontroller and an SX1262 onto a single silicon die. Using a more sophisticated FEC algorithm allows for improving the range through coding gain and also increasing the likelihood of successful packet capture. Coding gain is a metric that describes, typically in dB, the improvement in receive sensitivity due to the ability to correct bit errors in received data.

QMesh currently implements the Reed-Solomon-Viterbi (RSV) forward error correction algorithm. RSV concatenates a convolutional code inner code with a Reed-Solomon outer code. RSV FEC has been used in e.g. the Voyager spacecraft and is currently used with the telemetry frames on the AO-40

<sup>6</sup> RM0461, "STM32WLEx advanced Arm®-based 32-bit MCUs with sub-GHz radio solution". Available at [https://www.st.com/resource/en/reference\\_manual/dm00530369-stm32wllex-advanced-armbased-32bit-mcus-with-subghz-radio-solution-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/dm00530369-stm32wllex-advanced-armbased-32bit-mcus-with-subghz-radio-solution-stmicroelectronics.pdf) on July 11, 2020.

amateur satellite<sup>7</sup>. QMesh's implementation uses a (32,24) Reed-Solomon outer code with a ½ rate convolutional inner code with constraint length of 7. A bit-level “rectangular” interleaver, conceptually like the AO-40's FEC interleaver, spreads error bursts throughout the data packet.

RSV was chosen over more modern FEC algorithms such as LDPC, Turbo, and Polar codes for several reasons. First, the RSV algorithm has an easy-to-use, microcontroller implementation-friendly library implementation in `libcorrect`<sup>8</sup>. `libcorrect` is a simplified version of Phil Karn's `libfec`<sup>9</sup> implementation that removes many of the GNU/Linux (and x86)-specific aspects of the library. Likewise, there was not an easy-to-use, fast version of Turbo codes or Polar codes; moreover, practical implementations of polar codes appear to be encumbered by patents. LDPC codes, while having numerous, reasonable-to-use implementations, require the generation of a parity check matrix specific to the block size and number of parity bits.

While the AO-40's RSV FEC can offer up to 10-11dB of coding gain, it is expected that the amount of coding gain is significantly lower in QMesh due to using hard decoding (the LoRa radio does not provide soft decoding information in any sort of straightforward way) and due to the shorter message length. Soft decoding, where the receiver conveys its confidence in the received symbol's correctness to the FEC decoder, typically improves the coding gain by 1-2dB. Another issue reducing the realized coding gain is the short message size. Short messages can considerably reduce the coding gain by having fewer observations from which to correct messages. A very rough estimate of the RSV FEC in QMesh is that it provides 5-7dB of coding gain versus the built-in Hamming codes. While the author estimates a 5-7dB coding gain for the RSV implementation, this may underestimate the coding gain in certain situations, given that Haystack Technologies claims a 13dB performance gain using their FEC, which appears to use soft decoding and Polar codes<sup>10</sup>.

### **QMesh Hardware and Software Implementation**

A key enabler of QMesh is the massive improvement in the compute power and RAM/storage in commonly available, low-cost, low-power microcontrollers over the last decade. These microcontrollers allow for fine-grained, low-level, real-time control while running Real-Time Operating Systems (RTOSes) and supporting high-level C++ constructs like STL containers and smart pointers. These software constructs greatly improve programming productivity.

---

<sup>7</sup> James Miller G3RUH, “Oscar-40 FEC Telemetry”, Available at <https://www.amsat.org/amsat/articles/g3ruh/125.html>, accessed July 11, 2020.

<sup>8</sup> Brian Armstrong, “libcorrect”. Available at <https://github.com/quiet/libcorrect>, accessed July 11, 2020.

<sup>9</sup> Phil Karn, “Forward Error Correcting Codes”. Available <http://www.ka9q.net/code/fec/>, accessed July 11, 2020.

<sup>10</sup> “Say Hello to XR2”, Available <https://www.haystacktechnologies.com/xr-error-correction/>, accessed July 11, 2020.

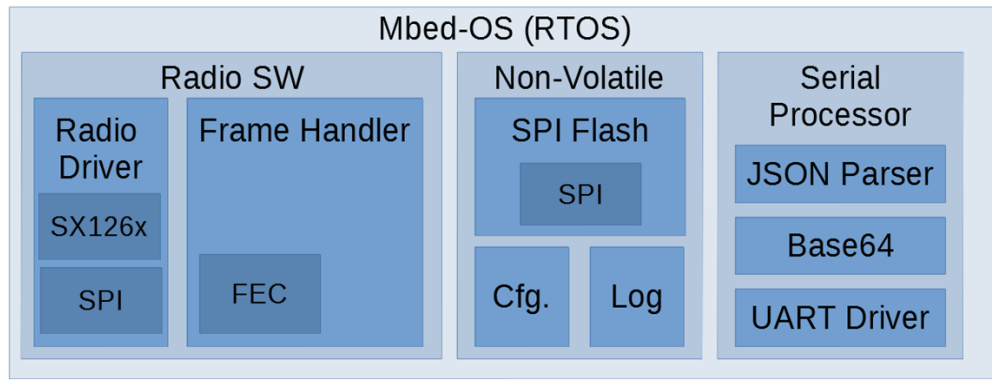


Figure 5: QMesh Software Architecture.

QMesh's current proof-of-concept hardware uses the STM32 NUCLEO-144 boards. ST designs the NUCLEO series of development boards to provide a “simple” development environment for their STM32 products. Non-MCU components are typically limited to a few LEDs and for some boards an Ethernet interface. The NUCLEO line comprises different form factors ranging from small boards that are similar to the Adafruit Feather form factor, all the way up to the larger NUCLEO-144 series, which provides a large number of I/O pins and uses the 144-pin STM32 MCUs. These MCU boards are relatively affordable, with the high-end STM32H743ZI2 costing roughly USD \$30 on Digi-Key as of the writing of this paper.

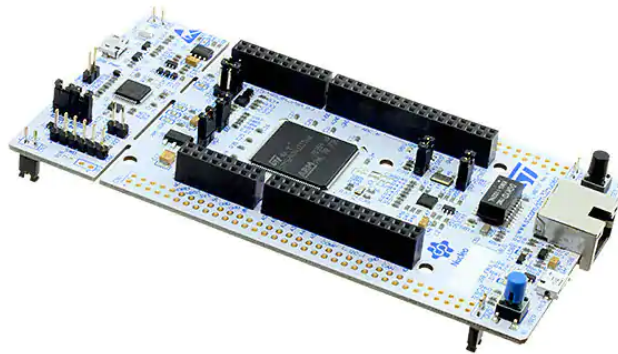


Figure 6: An STM32 NUCLEO-144 board.

The STM32H7 is the high-end line for STM32 microcontrollers. All H7s contain at least an ARM Cortex-M7, whose microarchitecture is like a Pentium MMX, and runs at 400-480MHz. Some variants also contain a secondary Cortex-M4. The STM32H743ZI contains an ARM Cortex-M7 running at 480MHz, 1MB of SRAM, and 2MB of flash.

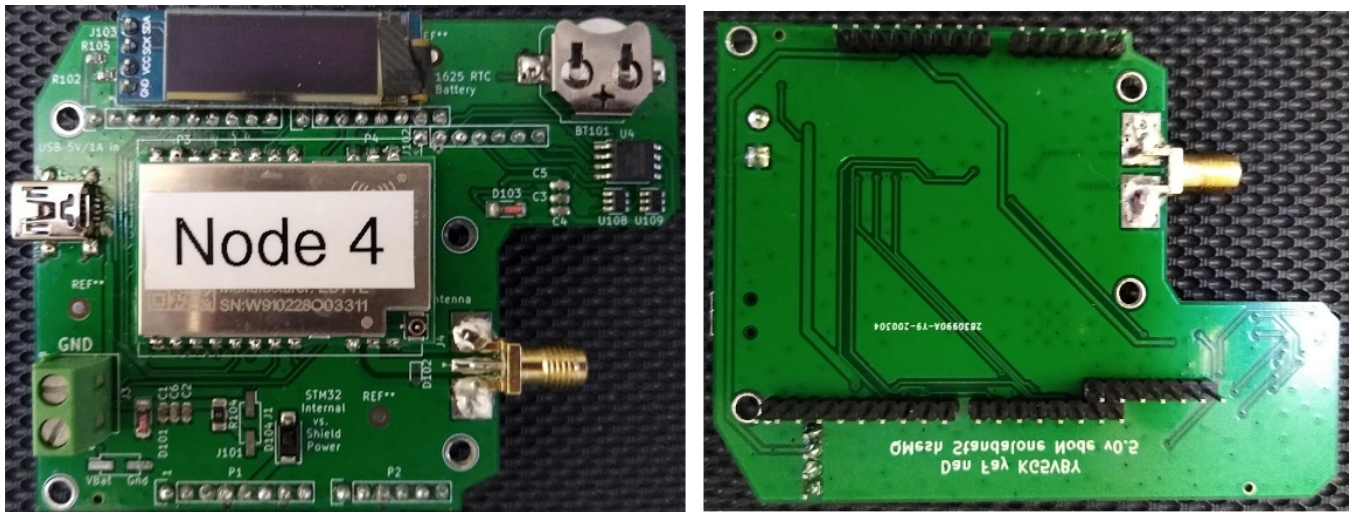


Figure 7: Closeup of the QMesh radio board.

The radio board is a custom design designed to utilize the “Zio” connector on the STM32 NUCLEO-144 boards. It is mostly compatible with the Arduino Shield form factor, save for some additional pins used to connect to the MCU’s QSPI bus. It is designed to power both itself as well as the STM32 board via a 5V source supplied via either screw terminals or a mini-USB-B connector. The combined power consumption for both the radio shield as well as the STM32 board stays below 1A, which allows it to be powered by common USB power sources.

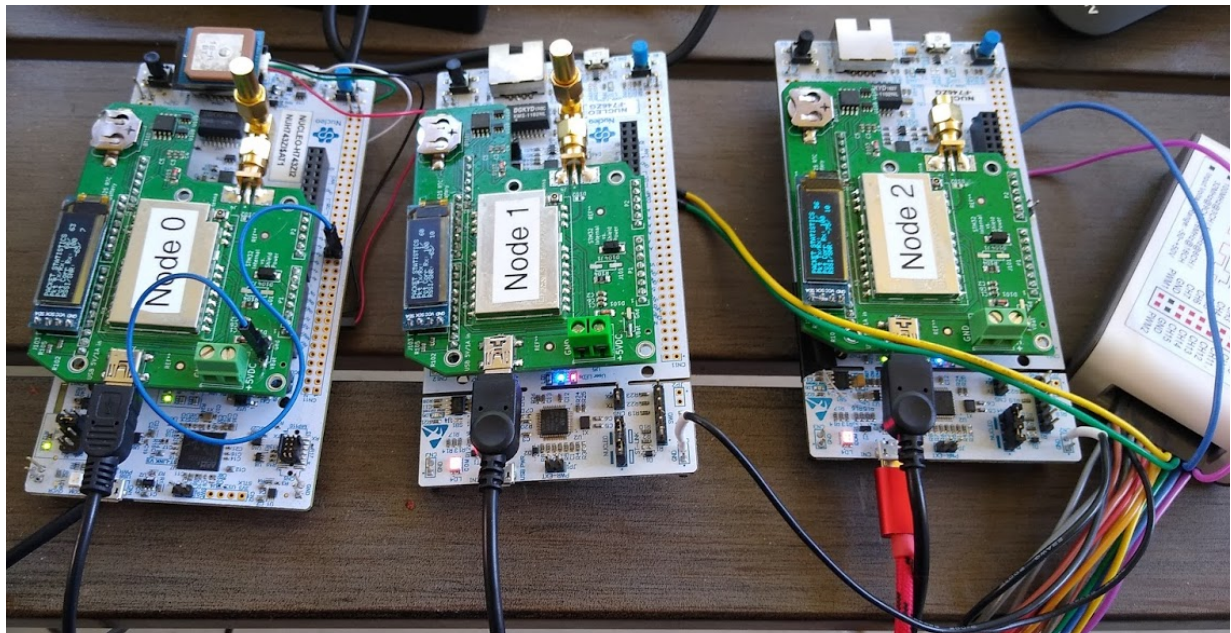


Figure 8: Nucleo-144 with radio shield.

At the heart of QMesh’s radio board is an EByte E22 module. While Chengdu EByte produces several different LoRa modules, these modules (the E22 400M30) have several useful features:

1. **1W transmit power.** This output level is sufficient to serve as an input into “DMR amplifiers”, allowing for higher transmit power if required or desired.
2. **External LNA.** This component can improve receive sensitivity by ~2dB.

3. **TCXO.** A TCXO improves frequency stability and accuracy which facilitates the frequency wobbling feature implemented by QMesh. It also enables the ultra-low-data-rate, narrow-bandwidth, high-sensitivity LoRa settings.
4. **SPI interface.** SPI interface enables the precise control of the LoRa modem that QMesh needs to implement its TDMA-based protocol.
5. **33cm and 70cm versions.** Two variants of the module support the 33cm and 70cm amateur bands.

In addition to the LoRa module, the radio board contains a 128Mbit Winbond QSPI NOR flash chip that is used for storing configuration and logging information and a 128x32 monochrome OLED display that can display live information such as packets received, SNR, RSSI, etc. The OLED display uses an SSD1306 controller chip and communicates with the STM32 microcontroller over I2C. Electrostatic discharge (ESD) and overvoltage protection has been added to the board as well through Transient Voltage Suppression (TVS) diodes. The board was designed in KiCAD and manufactured at JLCPCB. JLCPCB is a large board house in Shenzhen, China that can inexpensively manufacture the PCB itself as well as populate many of the board's components.

#### Approximate Bill of Materials for a QMesh “Standalone” (repeater) Node

Item	Purpose	Vendor	Cost (USD)
NUCLEO-144 STM32H743ZI2	MCU Board	Digi-Key	\$27.00
EByte E22 400M20	1W LoRa Module	AliExpress	\$10.50
Solar charger board	Charge Li-Ion battery, supply node	AliExpress	\$8.99
Solar panel 3W	Power	AliExpress	\$5.44
26650 Li-Ion Cell	Battery storage	AliExpress	\$4.49
26650 Holder	Battery holder	AliExpress	\$0.94
Retevis RHD-771	2m/70cm whip antenna	AliExpress	\$4.71
SMA Connector	SMA port for RF shield	AliExpress	\$0.21
SMA Cable	Connect RF board to antenna connector	AliExpress	\$1.42
SMA Bulkhead Connector	Provide weatherproof antenna connector through case	AliExpress	\$1.00
OLED Display	Live display of information	AliExpress	\$1.43
PCB Mfg. and assembly of RF board	Partially populated, assembled RF shield board	JLCPCB	\$5.82
Apache 1800 Case	Weatherproof Case	Harbor Freight	\$12.99

<b>Total</b>			\$84.94
--------------	--	--	---------

### **Experimental Setup**

The experimental setup presented here is an attempt to simulate a worst-case scenario for QMesh communications: where identical nodes transmit at under the exact same conditions: same location, same polarization, etc. The setup used here involves placing three 2m/70cm whip antennas a quarter wavelength (at 433MHz) from each other, as shown in Figure 6. This setup allows for testing collisions between two and three identical nodes. The worst-case scenario for QMesh is multiple transmitters transmitting at nearly the same power. To test this situation, a setup with three boards transmitting on three 2m/70cm whip antennas spaced one-quarter wavelength apart was used. Successful communications using this setup should adequately demonstrate the ability of QMesh to successfully capture packets amid large amounts of collisions.



*Figure 9: Setup used to test capture success.*

Testing multipath effects is highly challenging, as fading can quickly change by even small changes in location. For this early-phase testing, the packet reception was tested by the author walking around his backyard with a node and measuring the packet receive rate. Walking around the backyard should help provide a variety of multipath fading situations to test the ability of QMesh to withstand interference from two and three nodes simultaneously retransmitting.

### **Experimental Results**

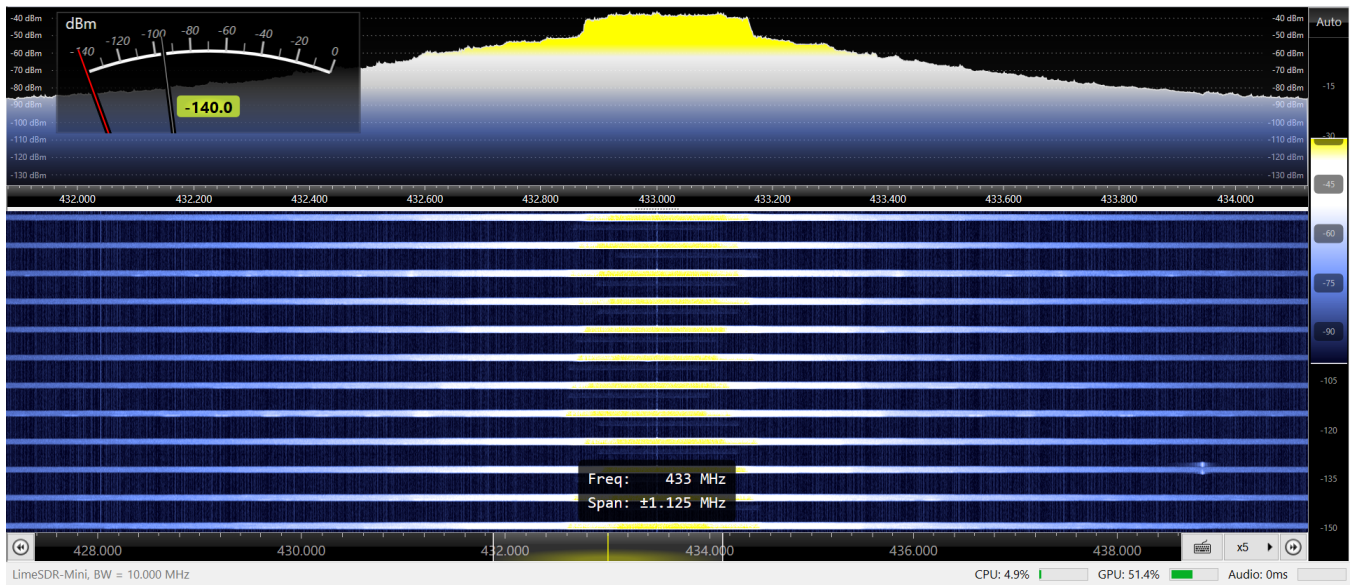


Figure 10: Waterfall of three QMesh nodes retransmitting a signal.

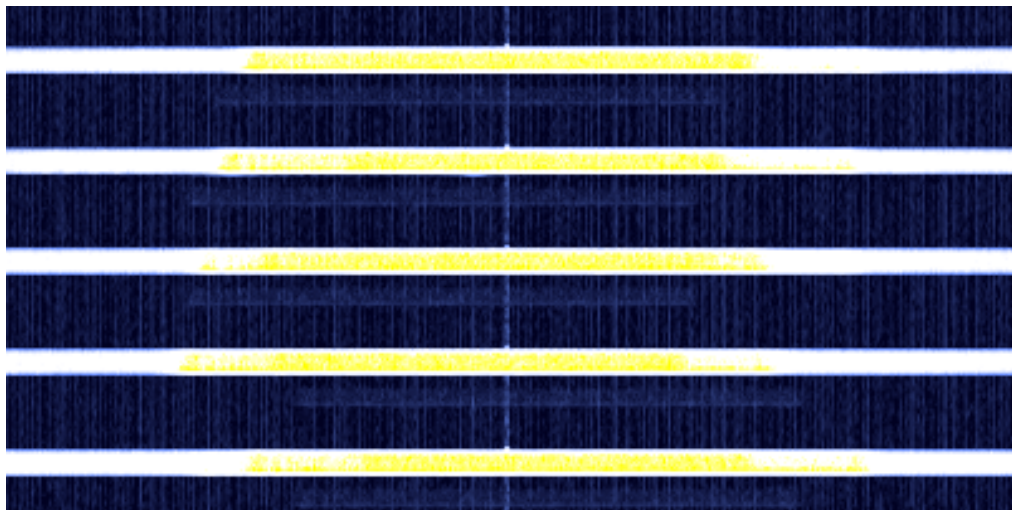


Figure 11: Several transmits enlarged. Visible are the varying transmit power levels as well as the (very faint) originating transmission.

The above figures show the waterfall for three nodes retransmitting. Three things can be observed here. First, the frequency wobbling is visible as a left-right shift in frequency between the chirp packets. Second, the variable transmit power can be seen as well, with the overlapping transmits creating lighter shades of yellow at the edges of the overlapping chirps. Finally, the original transmit is visible as a very faint LoRa packet.

In a single transmitting node configuration, FEC appears to “steepen” the packet error rate curve. Essentially, it causes the LoRa modem to either receive the packet correctly, or not receive anything at all -- there are considerably fewer cases where a received packet is incorrect. It also does appear to improve the receive floor by about 2-4dB. It is likely that the relatively modest improvement in the receive floor is due to the SX126X's receiver being tuned to only receive packets that are likely to be

error-free. Future work will look at using different detection parameters in the SX126X's Channel Activity Detect (CAD) feature<sup>11</sup> as a way of lowering the detection floor.

Packet reception appears to work well. When leveraging the combination of FEC, frequency wobbling, time offsets, and variable transmit power, a QMesh node can receive packets at a 99% PRR when two or three nodes are simultaneously retransmitting. Removing the variable transmit power does not appear to have a significant negative effect on the Packet Receive Rate (PRR), with a 99% PRR for both the two and three retransmitter situations. The final test looks at the PRR with no FEC. For this scenario, three nodes leads to a 90% PRR, two nodes a 93% PRR, and one node a PRR of 100%. The non-FEC-using tests strongly suggest that forward error correction is essential and highly beneficial for synchronized flooding scenarios. Without FEC, the PRR drops considerably from 100% to 93% for two repeating nodes and 90% for three repeating nodes.

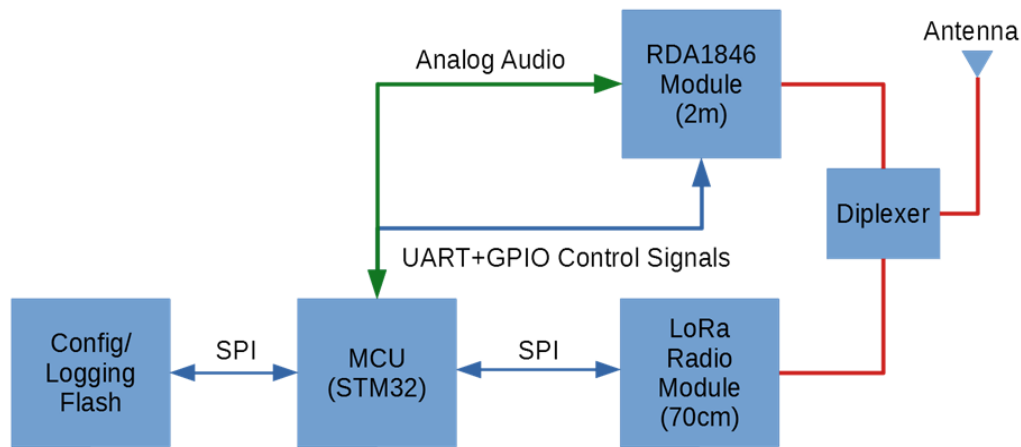
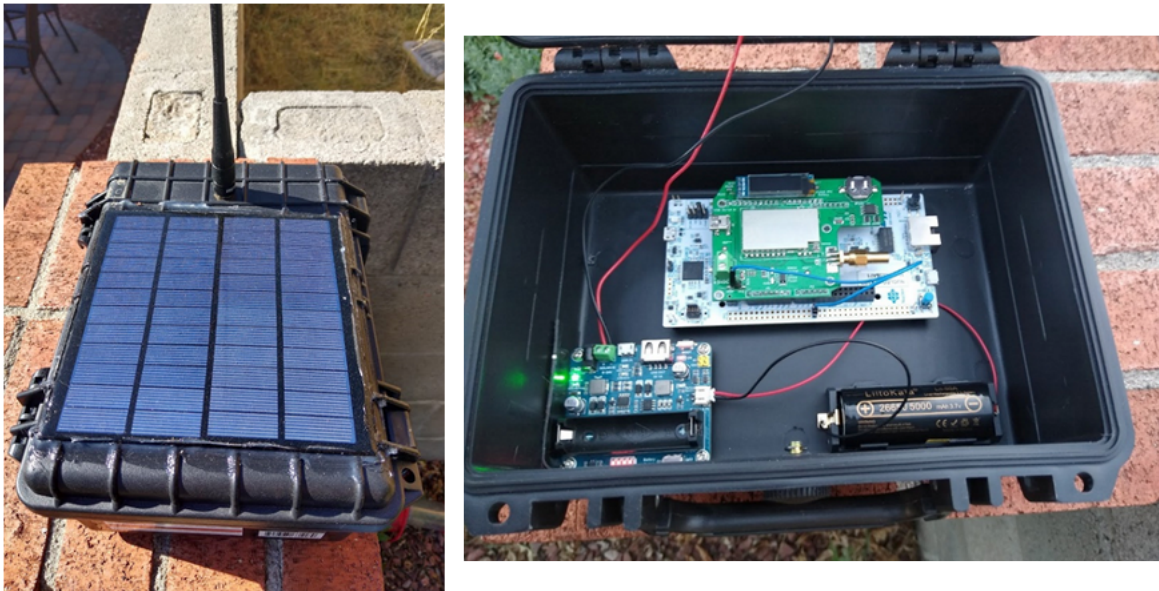


Figure 12: Proposed architecture of QMesh Analog Repeater

### Future Work

A medium-term use of QMesh is to develop a series of small repeaters that bridge narrowband analog FM communications with QMesh's digital LoRa mesh. The goal of this micro-repeater is to make QMesh as accessible to radio amateurs as possible by allowing for the use of existing FM radios. FM radios would communicate with QMesh repeaters in the same manner as they would with a standard FM repeater; the QMesh repeater would receive the FM voice, encode it as `codec2`-encoded voice, send it out digitally over QMesh, and the other repeaters would in turn decode the `codec2` voice and transmit it out as FM. This capability can also be used to bridge the AFSK-over-FM waveform typically used by APRS with the QMesh protocol. This setup is physically compact and low-power, enabling it to serve as a small, solar-powered self-linking repeater.

<sup>11</sup> "Application Note: SX126x CAD Performance Evaluation", <https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000Q1Ec/b29BFJzTZY420v0tql3kVC6nzHRQizlfWGjDfcC1Dyc>, accessed August 14, 2020.



*Figure 13: Mockup of a solar-powered QMesh standalone repeater.*

Enabling this feature are the existence of compact, low-cost, low-power RDA1846S modules that are essentially a “Baofeng-on-a-module”. The RDA1846/RDA1846S is an SDR-based narrowband FM chipset that contains nearly everything needed to implement a narrowband FM radio. The Dorji DRA818(U/V) is an example of this module, where Dorji combines the RDA1846S with a microcontroller to provide a simple, UART-based interface to command and control the RDA1846S chip. An alternate version with full-duplex FM communication capabilities would use two of the DRA818V modules, with either both on the same band (requiring a duplexer and/or separated antennas), or on 2m/70cm in a cross-band mode. For these setups, QMesh would operate on 33cm instead of 70cm.

One application being explored is to develop trackers for the America’s Cup gas balloon race that occurs every year at Albuquerque’s Balloon Fiesta. These trackers would collect position information via GPS and ultimately forward it to an APRS digipeater or iGate over 2m AFSK. Since there is limited APRS coverage in the interior regions of the United States, QMesh would serve to link the various competing balloons together so that one within range of an APRS digipeater or iGate would forward traffic from other balloons that are not. Using one of the low-datarate LoRa settings, the receive sensitivity should be high enough that some limited beyond-line-of-sight QMesh communications should be possible.

Other future work will examine other improvements to the QMesh protocol, such as channel access (i.e., CSMA-CA vs. ALOHA, etc.). Some physical-layer improvements could be explored, such as developing soft decoding of the LoRa modem, more advanced FEC algorithms such as Low-Density Parity Check (LDPC), and polar codes. Another physical-layer improvement to explore is receive-antenna diversity. It should be straightforward to implement a two-way selection diversity scheme with two QMesh standalone nodes.

### **Other Related Work**

There exist some other projects performing similar goals to the QMesh project. The M17 project<sup>12</sup> intends to replace various digital narrowband standards such as DMR, DStar, and C4FM with a fully open protocol (they all use the proprietary AMBE vocoder), including using the open source `codec2` vocoder. FreeDV<sup>13</sup> is a project that intends to supplant analog SSB voice with a digital voice protocol that combines `codec2`-vocoded digital voice with a custom OFDM waveform. A key goal is to develop a voice mode that can function at a lower SNR than SSB.

Multiple mesh network projects exist that serve to share APRS-like telemetry information between nodes. The most-successful commercial project is Gotenna Mesh<sup>14</sup>. This device works on the 915MHz/868MHz ISM bands to link together small dongles. These dongles interface with a smartphone via Bluetooth to provide location information as well as text messaging/chat. Meshtastic<sup>15</sup> provides similar functionality to Gotenna Mesh but is open source. It is designed to work with various common, open-source LoRa boards. Disaster Radio<sup>16</sup> is similar in concept, although it specifically targets stationary, solar-powered nodes spread throughout an area.

QMesh also draws from ideas used by New Packet Radio (NPR)<sup>17</sup>. NPR carries medium-speed IP traffic over 70cm. While not a mesh network, NPR inspired several ideas within QMesh. The first one is to use common, 1W ISM band RF modules for RF communications, instead of attempting to develop a custom waveform or RF board design. Doing so simplifies the design, lowers its cost, and lowers its power consumption as well. Second, QMesh shares the notion that so-called “DMR amplifiers” can be used to provide higher power outputs from these devices if necessary (in practice, a 1W input is sufficient to drive these amplifiers to an output of 20-25W). Finally, NPR demonstrated that STM32 microcontrollers can provide the precise timing needed for TDMA protocols.

Commercial projects which inspired QMesh include Virtual Extension,<sup>18</sup> which uses synchronized flooding for IoT devices, as well as the Bluetooth Mesh standard<sup>19</sup>, which uses flooding. Both protocols showed that flooded mesh networks are suitable for mesh network communications. Finally, Haystack<sup>20</sup> demonstrated that adding a Forward Error Correction algorithm more sophisticated than the built-in Hamming FEC can provide big performance improvements for LoRa.

---

<sup>12</sup> “M17 Project”, <https://m17project.org/>, accessed July 29, 2020.

<sup>13</sup> “FREEDV: OPEN SOURCE AMATEUR DIGITAL VOICE”, available <https://freedv.org/>, accessed August 15, 2020.

<sup>14</sup> “GoTenna Mesh”, <https://gotennamesh.com/products/mesh>, accessed July 30, 2020.

<sup>15</sup> “What is Meshtastic?”, <https://www.meshtastic.org/>, accessed July 30, 2020.

<sup>16</sup> “Disaster.radio: “, <https://disaster.radio/>, accessed July 30, 2020.

<sup>17</sup> “New Packet Radio”, <https://hackaday.io/project/164092-npr-new-packet-radio>, accessed July 31, 2020.

<sup>18</sup> “VEMesh IoT Products”, <https://virtual-extension.com/products/iot-products/>, accessed July 31, 2020.

<sup>19</sup> “Introducing Bluetooth Mesh Flooding”, <https://www.bluetooth.com/blog/introducing-bluetooth-mesh-networking/>, accessed July 31, 2020.

<sup>20</sup> “How to Quadruple the Range of LoRa”, <https://www.haystacktechnologies.com/2019/10/07/xrmode2/>, accessed July 31, 2020.

## **Summary and Conclusion**

This paper provided an overview of QMesh, which is a synchronized flooded wireless mesh networking protocol leveraging unique characteristics to allow for successfully receiving colliding packets. It examined several features of QMesh, including the use of sophisticated Forward Error Correction, randomized frequency offsets, and randomized timing offsets. The results presented here demonstrate that a LoRa-based synchronized flooding protocol can achieve high packet receive rates if Forward Error Correction is used.

Finally, this paper discussed the long-term goals of QMesh, which is to create a voice-capable mesh network. It showed a proposed hybrid analog FM/QMesh architecture that is compatible with existing narrowband FM radios, including the AFSK-over-FM waveform commonly used by APRS. This hybrid setup should make QMesh more accessible and useful immediately without requiring people to buy new radios. It also allows for bridging between analog FM repeater systems and/or APRS digipeaters/iGates.

*For more information about QMesh, please see the project's Github repository at <https://github.com/faydr/QMesh>. The author can be contacted at [daniel.fay@gmail.com](mailto:daniel.fay@gmail.com) and periodically posts about Amateur Radio and QMesh on the Twitter account [@faydrus](https://twitter.com/faydrus).*